Connecting OKI And SQI: One Small Piece Of Code, A Giant Leap For Reusing Learning Objects.

Stefaan Ternier, Ben Bosman, Erik Duval
Katholieke Universiteit Leuven
Leuven, Belgium
{stefaan.ternier,ben.bosman,erik.duval}@cs.kuleuven.be

Lorin Metzger, Mike Halm
LionShare Project
The Penn State University
University Park, Pennsylvania,
{lmetzger,mjh}@psu.edu

Scott Thorne, Jeff Kahn
Open Knowledge Initiative
Massachusetts Institute of Technology
Boston, Massachusetts
{thorne, jeffkahn}@mit.edu

Learning object repositories are infrastructures for storing learning resources. If they offer a critical mass of materials, they have the potential to facilitate teaching and learning by offering means to reuse and integrate existing materials. This paper discusses search interoperability efforts. By bridging the OKI Open Service Interface Definition (OSIDs) for repositories and Simple Query Interface (SQI), a CEN/ISSS standard for interconnecting search engines, two worlds of repositories are consolidated. The results of this work have already been implemented in two systems: ARIADNE and the LionShare Peer-to-Peer (P2P) environment

Introduction

One of the problems facing education is that creating content is expensive. Reusing existing components, which are often referred to as learning objects, addresses this problem. When content creators search for materials to reuse, they typically don't care about origin. When searching for materials, creators want results that represent the best quality materials that satisfy their query. This paper thus focuses on providing interoperability between search engines and interconnecting repositories transparently for the end user.

When interoperability is discussed in this paper, it refers to the ability for disparate systems to query the resources of another system and to return results in a standardized manner and vice versa. Thus, the goal of this project was to implement the OKI Repository OSID and SQI (Simon et al. 2005) as a mechanism to demonstrate interoperability between LionShare and ARIADNE repositories. In the case of LionShare and ARIADNE, these two environments are very different in design. LionShare is a completely distributed system, P2P network comprised of peers that have the ability to act as both clients and servers to other peers on the network. The ARIADNE Knowledge Pool System (Duval 1999), on the other hand, is a collection of servers accessed, in the classic, client-server model, where an entity is either client or server but not both.

Simple Query Interface

The Simple Query Interface (SQI) is an open, accredited standard widely implemented in organisations like EUN or networks like Prolearn or GLOBE. SQI specifies how to transport a query to a server. The standard is agnostic in terms of implementation and was therefore specified in an abstract manner. When a concrete SQI implementation is created for a repository, it is necessary to provide a binding in a given environment. As most of the stakeholders in SQI were interested in providing remote access to their repository, the SOAP web services binding of SQI is currently the most popular. Creating a concrete binding for a search service like SQI is analogous to creating a binding for a metadata standard. As an example, the IEEE LOM standard is a specification that describes the semantics of different metadata fields relevant for education. Without a binding, like XML or RDF, the standard is impossible to implement and useless for interoperability.

SQI depends on tokens referred to as sessions. Therefore, the specification comes with an optional session management specification. Within the session management specification, two methods enable the creation of a session and another method is used to destroying them. These tokens offer a great deal of flexibility to an application. For example, these token enable the interface (SQI) with a flexible mechanism for authentication by binding the credentials to the token. Secondly, a system can use these tokens to store a search profile on the server. And finally, tokens can also be used to track usage of the search service.

SQI is applicable in many search scenarios as the specification makes clear distinction between synchronous and asynchronous responses. In the synchronous scenario, a client invokes the method *synchronousQuery* that returns the first 25 results. The *asynchronousQuery* method handles scenarios where a client does not want to be blocked waiting for a server to respond. An asynchronous query is executed in three steps:

1) The client informs the server where to send its results. It uses the method *setSourceLocation* to transport the location of its listener service. Note: one can make this step redundant by attaching this location to the search profile that is identified by a token.
2) The client sends its query to the target.
3) The target sends the results asynchronously to the listener service implemented by the client. This step requires the client to be reachable by the server.

OKI Repository OSID

The Open Knowledge Initiative (O.K.I.) is a MIT-lead, community effort to improve interoperability among applications and the enterprise systems services on which they depend. O.K.I. focuses on specifying the contracts between service consumers and providers. Open Services Interface Definitions (OSIDs) are well-defined integration boundaries that are neutral with regard to programming language and implementation detail. There are OSIDs for common services such as authentication, authorization, hierarchy, repository, scheduling, and workflow and eLearning services such as assessment, grading, course management, and repository.

The Repository OSID describes generic methods for searching, accessing, and updating content, including discovery of the metadata structures. This allows programs to concentrate on their function and interaction with the user, while on the other side of the software contract a repository "plug-in" or driver handles the details of how this is achieved with a particular repository. Figure 1 illustrates how an LMS like HarvestRoad can communicate through this kind of drivers with different repositories.

Separating the front end application from the back-end content store in a regular way is particularly beneficial in the area of Repositories, since it enables front-end tools to be easily integrated with many content sources. Conversely it enables content stores to be accessible from many different applications. By agreeing on an integration point ahead of time, applications can be produced and distributed that can then take advantage of new content sources as particular drivers for these are created. This is analogous to print drivers; the application knows how to request printing in a general way, and another piece of software, the print driver, handles the details of communicating with a particular printer.

How to provide a complete query service

Although the main objective of SQI and OKI is providing interoperability, neither are complete solutions. More than a query service is necessary to make a search client and a repository interoperable. As an example, both initiatives are agnostic about the query language that can be used. While this is clearly a good design decision, it implies that both client and server agree on a common query language in order to be interoperable.

Some specifications make this information available through the services they offer. As an example SRW offers an explain method. Other initiatives store this information in a third party system like UDDI registry.

Customizing SQI

SQI builds on other standards and specifications. A complete SQI implementation contains a query language and a binding for that language. Furthermore, it specifies how session management can be interpreted. These components are needed to complete a SQI implementation.

First, an implementer needs to consider which query languages are to be supported. Currently, most targets support VSQL -- a proprietary query language that is metadata standard agnostic. This format only enables the representation of a list of search terms and it requires the repository to provide a meaningful mapping to its metadata. Apart from VSQL, a developer can add support for an arbitrary number of other query languages.

SQI is agnostic about the metadata standard and the binding to transport results to the user. Making a decision about this metadata standard binding, rests partially on the choice of a query language itself. For a query language that returns variable bindings, like XQuery or QEL, an implementer must support the binding specified by these languages. For query languages that result in hits that match a query, an implementer has some freedom in choosing an appropriate results format. In the SQI world, most repositories agree on the LOM XML binding as a results format. Nothing prevents also (or only) providing RDF Dublin core formatted metadata besides LOM XML.

Finally, each method in SQI needs a token (a session identifier). This token can either be requested through a service (e.g. the session management service that comes with SQI) or a developer can opt to use persistent tokens. Persistent tokens can be used to store a search profile on a target allowing a (group of) search clients to rely on a stored configuration. As an example, a client could receive a token that enables searching in graphical materials only.

SQI was designed such that making this information available, is not part of SQI. The federated search engine that comes with the ARIADNE toolset fetches this information from a UDDI registry where both GLOBE and Prolearn partners can register their services. Together with the location of the services, this registry also stores the metadata that describes them. Based on this information, a search client can limit itself to infrastructures it is interoperable with.

Customizing OKI

OSIDs concentrate only on the specifics of how an application calls a type of service and remains neutral on the specifics of how the service is provided. There are other types of specifications that deal with other areas, such as what data formats are used and how information flows between machines. OSIDs are meant to complement these other types of specifications, so that they can be used together.

OKI is a Service Oriented Architecture. There are service definitions for many functional areas. While each service is abstract, there is a place where specific protocol, algorithm, or data format detail is indicated. For example, while the general search method for a repository accepts a Type argument that can hold any value, a specific implementation will support only a particular set of values such as a Title search, an Author search, and so on. Another area for Types in the Repository OSID is in Assets. While the interfaces include a general Asset Type, an implementation will make this specific, for example returning an Image Asset. Exactly what an Image Asset entails is specified in public document created by a community.

Bridging

Providing users of one repository access to another institutions repository is a great service. Access to resources grows exponentially as new repositories provide an interface for search. For example, suppose two repositories offer respectively *n* and *m* learning objects, by providing an interoperability interface on top of one another the search space suddenly becomes *n+m* learning objects. This kind of interoperability becomes even more exciting when bridging to a group of repositories. This section will therefore explore the possibilities of bridging both SQI and OKI. Creating these bridges depends heavily on implementation decisions, discussed in the previous chapter.

Although SQI and OKI are both abstract and implementation language neutral specifications, practice shows that only the SQI web services binding is commonly used and that OKI adapters are commonly used as adapters pluggable into an existing software infrastructure. It is then up to the adaptor to use some kind of network protocol to connect to the repository application. This makes OKI and SQI to a certain extent complementary. Used in the right context, SQI and OKI can be connected to provide a high level of interoperability between existing learning object repositories and the tools needed by educators and researchers to locate and retrieve learning objects from an array of repositories.

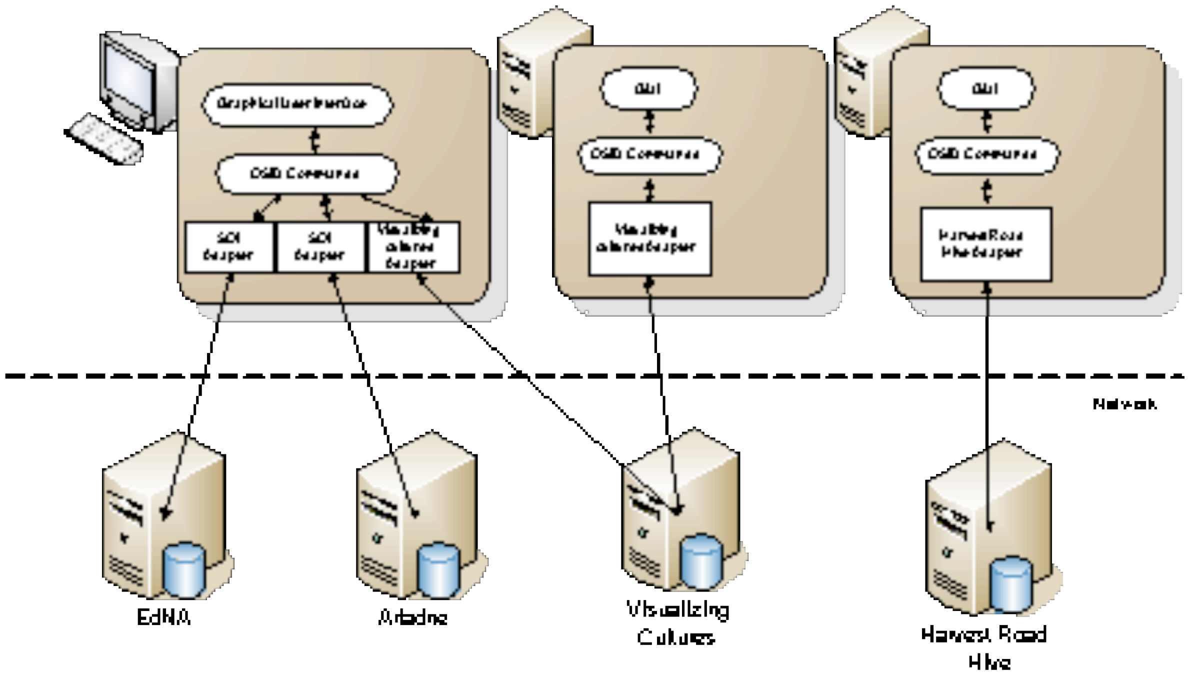

Figure 1: An OKI to SQI bridge

To achieve this level of connectivity, two bridges need to be constructed. The first bridge (fig. 1) provides OKI-enabled search and retrieval tools access to SQI target repositories. This simple bridge provides all existing OKI-enabled search and retrieval tools (i.e. LionShare, Learn eXact Package, HarvestRoad Hive Explorer, Tufts VUE, Sakai Twin Peaks and SearchParty), the ability to search content in SQI repositories.
From the previous section we learn that both SQI and OKI build on top of other specifications. For the bridge that was created here, a keyword-based search type that is supported by many OKI OSID implementations was mapped to VSQL.

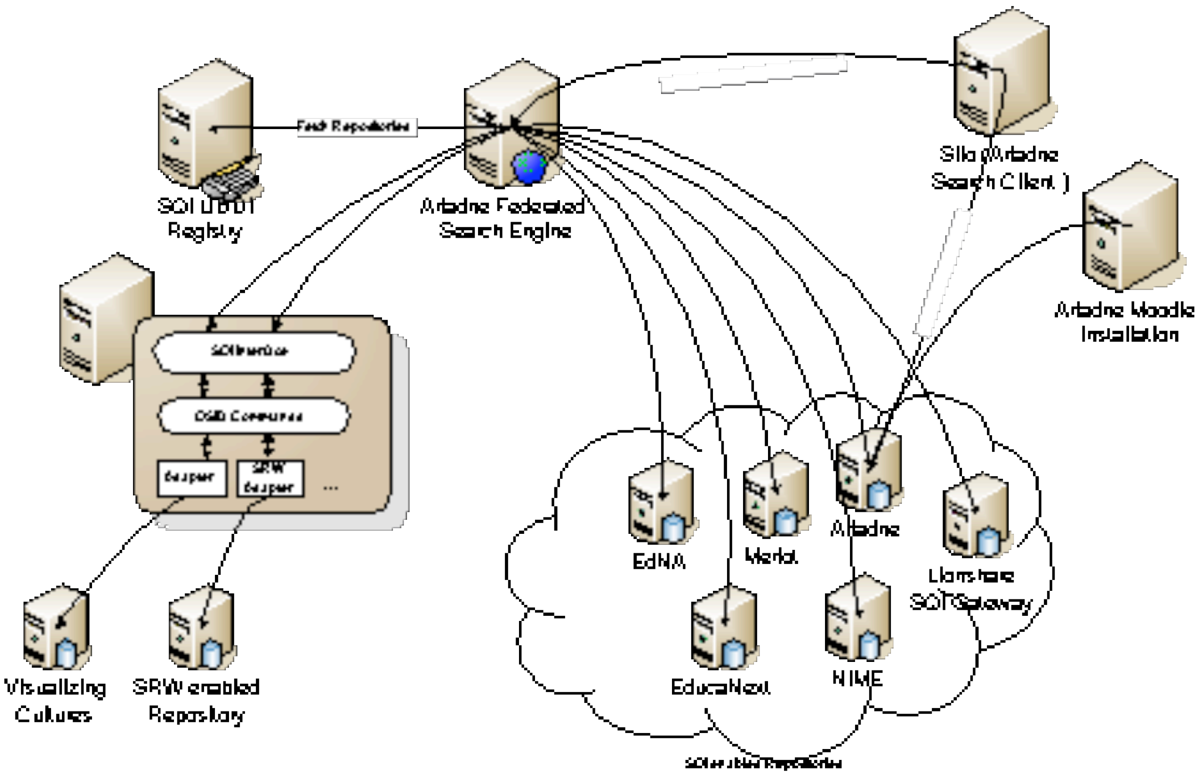

Figure 2: an SQI to OKI bridge

The second bridge (fig. 2) connects existing OKI-enabled repositories to the SQI enabled repositories. This can be achieved as an OKI OSID does not imply rules on how an adapter communicates to a back-end repository. For this purpose, a software module was created that implements the SQI web services binding. This module is capable of translating incoming SQI requests to OKI calls. Again queries and results are properly mapped into one another. Taking these mappings into account, SQI enabled search tools like Silo -- the ARIADNE search and indexation tool -- can transparently search repository OSIDs like MIT's visualizing cultures Image Database (VCID). Other repository content for which Repository OSID implementations exists include: selections from the Boston Museum of Fine Arts, The Tufts Digital Library, and the Rotch Visual Collections, at MIT. Many content sources such as JStor are working with the OSID in a pre-production environment.

These bridges have currently been implemented and tested as part of the ARIADNE and LionShare projects.

Current implementations

Lionshare

LionShare is a secure peer-to-peer (P2P) learning object distribution application. Its primary goal is to facilitate the distribution of localized content found on the personal computers of educators and researchers not having an easy way to publish this content to popular learning object repositories. The advantages of LionShare over the other P2P file sharing applications are its tight integration with centralized institutional authentication systems. This added security allows educators, researchers and students to confidently distribute content, because the digital identity of the person sharing the object is clearly identified in the search results.

Acknowledging there is extremely valuable content on centralized learning object repositories, LionShare includes the ability to search these repositories in conjunction with the personalized content available on the secure P2P network. LionShare achieves its repository searching capability by using the OKI Repository OSID, described above. Using OKI provides a standard interface, which makes it easy for the LionShare P2P client to "plug and play" with a variety of learning object repositories utilizing differing search and retrieval standards.

Another exciting aspect to LionShare is a new SQI to LionShare gateway. This SQI target allows users to treat the LionShare's P2P network collectively as just another big and very distributed learning object repository. When a LionShare user queries the P2P network, the query is sent to neighbouring peers. These, in turn, distribute the query recursively. A result of this behaviour is a peer that has issued or forwards a query, never knows when all the peers have returned their results. This usually provides query results lists that are updated as more peers respond asynchronously. This has motivated the implementation of an asynchronous SQI target.

ARIADNE

ARIADNE offers an SQI enabled Learning Content Management System that comes with an SQI target. As ARIADNE puts more emphasis on reusing materials, ARIADNE tools come with transparent access to other learning repositories. At a technical level, a federated search engine exposes itself to an SQI-enabled query tool becoming a virtual repository for learning content. After receiving a query through the SQI gateway, the query is federated to a set of SQI targets fetched from a UDDI registry (fig. 2). Note that this federated search engine needs SQI at two levels. First, it needs a SQI interface for the federated search engine to make the search tool useful in other contexts and second, the federated search tool needs the federated repositories to be SQI-compliant. From the previous information, we discussed interoperability requires a common set of standards. The federated search engine will therefore, only select interoperable SQI targets. Thus, the target repository needs to support VSQL as query language and LOM XML as results format. Currently, this engine searches in the following repositories: NIME (National Institute of Multimedia Education Japan), MERLOT, EdNa, LionShare, Citeseer, Educanext and VCID. Other implementations are ongoing.

Related Work

Search/Retrieve Web Service (SRW) is Library of Congress initiative that focuses on making libraries interoperable. SRW implementations usually come with CQL as query language, returning results in the Dublin Core format. An OKI repository OSID adapter has already been implemented. Through the bridge discussed in this paper, this world is now also available for SQI enabled repositories, provided that the adapter is added to bridge (fig. 2)

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) specifies a protocol for services that enable harvesting the metadata of a repository. A harvesting interface provides means to extract metadata from a

repository. Note that this is not the same as a service for searching. In order to provide interoperability with this world, an implementation that harvests from OAI-PMH and stores the metadata in an SQI enabled repository is ongoing.
A general implementation of the Repository OSID for an OAI service is available. The implementation has configuration options for the Base URL, mapping Sets to Repositories, the metadata prefix to use, and will search harvested results for a subset that matches some criteria.

In 2003, the IMS Global Learning Consortium released their digital repositories specification. This document makes recommendations on the kind of services a repository should implement. Specifications like the eduSource Communication Layer (ECL) later adopted these recommendations. An implementation of the Repository OSID for ECL is in final review. Modifications for the OSID's are managed under IMS specification maintenance process.

Conclusion

One theme of OKI and SQI in general is to survive technology change to provide investment protection for the development of user-facing applications. Moving transport or other specific technical detail out of the client makes it smaller and easier to maintain. This allows existing content to be transparently used in new clients and hence increases the Return on Investment in those content sources.

Another result of this work is that for a repository, it doesn't matter what specification is implemented. For organisations, it becomes hence more attractive now to implement a standard. Through bridges, like the ones presented in this paper, their repository can be plugged into virtually every architecture.

References

[1] 1484.12.1 IEEE Standard for Learning Object Metadata, http://ltsc.ieee.org/wg12.
[2] Ariadne. *http://www.ariadne-eu.org*
[3] Dublin Core Metadata Initiative, http://dublincore.org/.
[4] European schoolnet, http://www.eun.org/
[5] Globe, http://globe.edna.edu.au/
[6] HarvestRoad. http://www.harvestroad.com
[7] IMS DRI (2003) IMS Digital Repositories Interoperability - Core Functions Information Model, Version 1, http://www.imsglobal.org/digitalrepositories/index.html
[8] Lionshare P2P project. *http://lionshare.its.psu.edu*
[9] MIT Academic Computing. *http://blackshipsandsamurai.com*
[10] OAI. The Open Archives Initiative. *http://www.openarchives.org*
[11] OKI. http://www.okiproject.org
[12] Open Archives Initiative-Protocol for Metadata Harvesting, http://www.openarchives.org.
[13] Search/Retrieve Web Service, http://www.loc.gov/z3950/agency/zing/srw/
[14] Simple Query Interface. *http://www.prolearn-project.org/lori*
[15] E. Duval, E. Forte, K. Cardinaels, B. Verhoeven, R. V. Durm, K. Hendrikx, M.W. Forte, N. Ebel, M. Macowicz, K. Warkentyne, and F. Haenni. (1999) The ariadne knowledge pool system. Communications of the ACM, 44(5):72–78.
[16] B. Simon, D. Massart, F. van Assche, S. Ternier, E. Duval, S. Brantner, D. Olmedilla, Z. Miklós. (2005) A Simple Query Interface for Interoperable Learning Repositories. Workshop on Interoperability of Web-Based Educational Systems in conjunction with 14th International World Wide Web Conference (WWW'05). Chiba, Japan